Benjamin O'Neill

Efficient Asymmetric Cryptography for RFID Access Control

Computer Science Tripos - Part II

Emmanuel College

2018

Proforma

Name:	Benjamin O'Neill
College:	Emmanuel College
Project Title:	Efficient Asymmetric Cryptography for RFID Access Control
Examination:	Computer Science Tripos Part II - 2018
Word count:	TODO
Project Originator:	Dr Markus Kuhn
Project Supervisor:	Dr Markus Kuhn

Original aims of the project

TODO: Summarise project proposal.

Work completed

TODO:

Special difficulties faced

TODO: Maybe mention library incompatibilities, broken cards.

Declaration of Originality

I, Benjamin O'Neill of Emmanuel College, being a candidate for Part II of the Computer Science Tripos, hereby declare that this dissertation and the work described in it are my own work, unaided except as may be specified below, and that the dissertation does not contain material that has already been used to any substantial extent for a comparable purpose.

Signed, Benjamin O'Neill

Date: TODO

Contents

1 Introduction

TODO

The goal of my project is to implement a smartcard access control system based on asymmetric cryptography, that would be a suitable hypothetical replacement to the smartcard system deployed by the University at the time of writing, namely the MIFARE classic system. The proposed system shall address the various problems of the MIFARE classic system, as well as more general problems with symmetric access control protocols.

A working group...

There is currently a notable deficit of publically available smartcard systems based on asymmetric cryptography. Consequently, it would be worthwhile to produce a working open-source implementation that can be used to deploy practical systems.

As I will explain in the Project Outcome subsection (TODO: link), the project was successful, and the design requirements met.

1.1 Smartcards

A smartcard is a plastic card with an integrated RFID chip with some processing power. They are typically deployed in systems that require secure personal identification and/or authentication, including access control systems such as the university door system, and payment systems such as Visa bank cards. Possible applications also extend to general data storage.

The majority of smartcards conform to the standards ISO7810 (which defines most of the physical characteristics) and ISO7816 (which mostly defines the electrical characteristics and communication protocols). Aspects of contactless smartcards are defined in ISO14443.

Smartcards are passively powered by the terminal. They come in two varieties, contact cards and contactless cards. Contact smartcards have a gold-plated contact pads that allow electrical connection with the reader, whereas contactless smartcards are powered by Radio-Frequency Induction as specified in ISO14443-2. Some cards are dual-interface cards that support both. Dual-interface cards were used for the project, although only the contactless functionality was used.

Smartcard-related details more relevant to the implementation of the project are outlined in the Preparation section.

1.2 Asymmetric vs Symmetric Cryptography

The majority of smartcard systems in existence depend on symmetric cryptography. Under this system, the smartcard would be issued with one or many secret keys, which can be used to access various door terminals which also contain one of the keys the card was provided. This approach has a few problems which can be addressed with the use of asymmetric cryptography.

The memory available to a smartcard is limited, which limits the number of keys a card can maintain. In large organisations, this can lead to sloppy reuse of keys, often with most of the doors using the same key. A private key that is stolen from one of the doors or cards, or the key issuing program, could then be used to gain access to all the doors.

Asymmetric cryptography mitigates this risk by not requiring any terminals store secrets, and assigning each card its own key pair. It would allow doors to be configured to allow or deny individual cards, allowing for a much more fine-grained access control system that's robust against the loss of a key.

1.3 MIFARE classic problems

The MIFARE classic smartcard system is the system currently deployed by the university. It has been proven to be weak to many forms of attack.

-

1.4 Project Outcome

Overall, the project was successful, in that the success criteria have been met, and the project was completed broadly to the anticipated timetable. However, there was not sufficient time to implement any of the optional extensions suggested in the project proposal.

The primary difference between the initial project proposal and the course that the project took in reality was that in the project proposal it was suggested that there will be a full and proper comparison of available protocols. In practice, many alternative protocols were looked into briefly but it would have been difficult to accurately assess any of them without a much lengthier investigation likely involving real implementations. There simply wasn't time.

2 Preparation

Might be better to have slightly flatter hierarchy here.

TODO Work done to refine proposal

2.1 Requirements analysis

List and explanations of success criteria.

- **The time taken for the reader to correctly authenticate the card should not exceed one second.**
  Smartcards have slow processing speeds, but the authentication process should be quick enough that it does not cause the user much inconvenience.
- **Access rights of a card should be revocable without having to modify the card itself.**
  As previously covered, this isn't possible with many existing systems.
- **The system should be sufficiently flexible to allow for unlimited doors, each with different access privileges.**
  Asymmetric cryptography will allow a system to easily meet this criterion. Each card need only maintain its own key pair, so the storage requirements won't increase with the number of encountered card readers.
- **The system should be secure**
  To meet this criterion, it is important to consider all kinds of attack. My research into problems with the MIFARE classic system and into other common smartcard vulnerabilities would inform the process of developing a more secure system.
- **The estimated large-scale introduction cost should not exceed the typical worst-case cost of a smartcard authentication system.**
  This criterion should be met as long as the proposed system doesn't prescribe additional expensive equipment beyond the smartcards themselves, the authentication terminals, and a card issuing system.

With the aim of proposing a suitable replacement, suggest introductions strategy.

Elaborate on project proposal. Point out important bits. Demonstrate professional approach.

2.2 Preparation strategy outline

Reference software engineering techniques, both here and elsewhere.

2.3 ISO7816

get background understanding,

2.3.1 APDUs

2.3.2 Other

APDU command/response structure, defined by ISO7816 etc.

2.4 Asymmetric Crypto

2.5 Current uni smartcards

Detail security flaws in MIFARE classic. Uni may be moving to MIFARE plus.
Give other shortcomings. (working group in university etc)

2.6 Identify tools

Better subsection name needed.

2.6.1 Java card

Information about Java Card i.e. what it is, different versions, how to program for it etc.
Initially bought 2.2.2. (Later bought 3.0.4) most recent 3.0.5 but physical card not available yet.
Descriptions on Oracle website
Differing development methods for different versions. (eclipse plugin for 3.0.5 most common, ant/command line more common in older ones. Used both ANT and command line when appropriate with 3.0.4).

Leave some more specific descriptions (especially relating to actual JC program structure) for the implementation section.

2.6.2 Card readers

Deets about card readers.

Analysis of development tools:

2.6.3 ANT

etc...

2.6.4 GPShell

2.7 Development

Getting familiar with development procedure. Example of test code and other such details.

had to learn various quirks and limitations of restricted Java Card language (e.g. int-to-short casting, restrictions on APDU passing, different garbage collection system). Very little online help about any of this.

3 Implementation

Could add more things in here relating to learning dev process.
Outline overall implementation strategy first?

3.1 Protocols

Decided what to implement.
Research into publicly available protocols, why I chose OPACITY, weaknesses initially identified etc.
Proceeded to research details in OPACITY e.g. EC Crypto.

3.1.1 EC Crypto

brief description of EC Cryptography.

3.1.2 ASN1

3.1.3 Other

(e.g. CMAC, whatever else)

3.2 Libraries

3.2.1 Host-side libraries

Rubenesque, cryptography, asn1. Elaborate of permissions.

3.2.2 Card-side libraries

3.0.4 library, JCMathLib used a bit (check if any of it actually made it into the final project).

3.3 Core implementation

3.3.1 Smartcard code

code produced (i.e. non-JCMathLib Java Card code including CMAC and ECDSA. (TODO: implement ECDSA)

Explain overall design decisions about how it was implemented.
Give examples of good code abstractions etc.

Illustrate similarities with and differences with regular Java code. Limitations in how APDU is passed, lack of int, standard libraries inaccessible, different libraries exist.

3.3.2 Host code

Explain how it was actually implemented.
Explain what would be different in an actual implementation.

3.4 Optimisations

Implemented PB. XML mock PB store on host. Card stores PB entries as a linked list. Describe PB optimisation.

Other optimisations that drastically improved timings. Only give overall time here, leave timing tables to evaluation.

3.5 Overall design

How card is issued (describe issuing), general holistic view of current system. (improvements described later in evaluation)
I think Issuer gets ID during issuing stage, adds it to allowed ID list in allowed terminals.
TODO: Is this suitable?

Work in the following: Could move a few things to here depending on whether learning the dev process counts as preparation or implementation.
Can highlight major milestones in the process.
APDU trace

TODO: Still need to implement a decent issuing interface.

4 Evaluation

4.1 Assessment of success

How well does it meet requirements, include data like timing tables etc. (should I do before/after optimisations here or in implementation? Maybe reference tables in implementation, or have references in implementation to tables here).

Security analysis would be needed here, considering security of the protocol itself, and other things e.g. whether issuing process has faults, whether PB entries can be altered maliciously etc.
Mention security analysis of OPACITY. It doesn't specifically talk about ZKM because it views the lack of terminal authentication, but mention claims about FS that apply.

Give sample output, perhaps photographs of current system components with discussion of how it would be implemented in reality.

Do costing etc

Any residual bugs?

How, specifically, does it address problems with MIFARE?

Properly test with multiple cards, multiple terminals. (simulate multiple terminals using multiple separate instances of the same script)

Look at philosophy of evaluation in HCI and Software Engineering courses.

Overall, Were original goals achieved?

4.2 Breakdown of results

Analysis of how much time each operation actually takes. May have already been covered in implementation but redundancy doesn't hurt. Outline which parts may in theory be improved.

TODO
Assessors look for signs of success, thorough and systematic evaluation. (see pink book 8.3)

5 Conclusions

No new info here, just summarise. Refer to introduction.

5.1 Achievements

Summarise the success criteria met and how viable the system is.

5.2 Knowledge Gained

Briefly outline what I have learned about which areas. Lots about smartcards, specific crypto functions like CMAC, asn encoding, EC cryptography...
How would I tackle a project differently in the future, and retrospectively is there a better way to have done it?

5.3 Future work

What general shortcomings does it have?
How to make more secure, faster. (library CMAC function would probably improve speed)

6 Bibliography

TODO - should link from main dissertation.

7 Appendices

TODO
Sample code, protocol diagram and other details, other details about the workings of smartcards e.g. from ISO7816, Java Card documentation.

8 Index

TODO (Optional)